

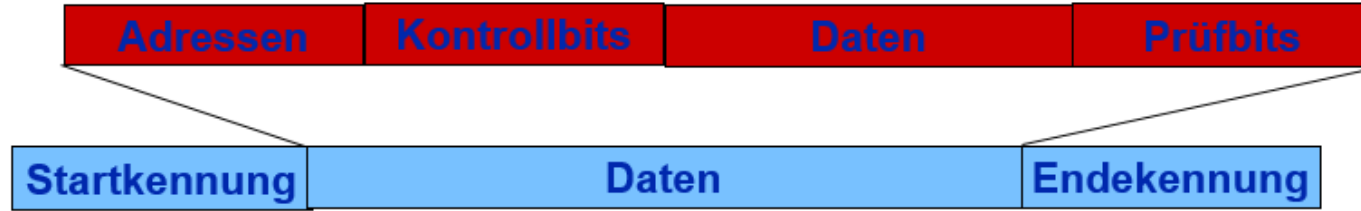
UbiComp – Teil 6: Netzwerktechnik und industrielle Kommunikation III

Prof. Dr.-Ing. Dorothea Schwung

Lernziele Teil 6

1. Sie sind mit der synchronen und asynchronen Datenübertragung vertraut.
2. Sie kennen das Konzept des Framings.
3. Sie können die unterschiedlichen Buszugriffsverfahren klassifizieren und die einzelnen Vertreter benennen.

Synchronisation & Framing

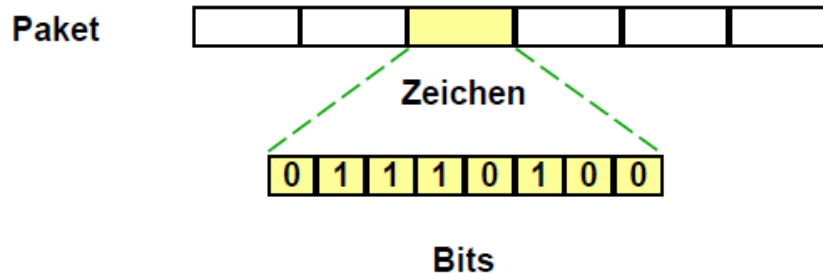


Schicht 2: Adressen für Sender und Empfänger
Dienstkennzeichen
Daten (aus darüber liegenden Schichten)
Sicherungs- und Prüfdaten (z.B. CRC)

Schicht 1: Start- und Endekennung

Synchronisation & Framing

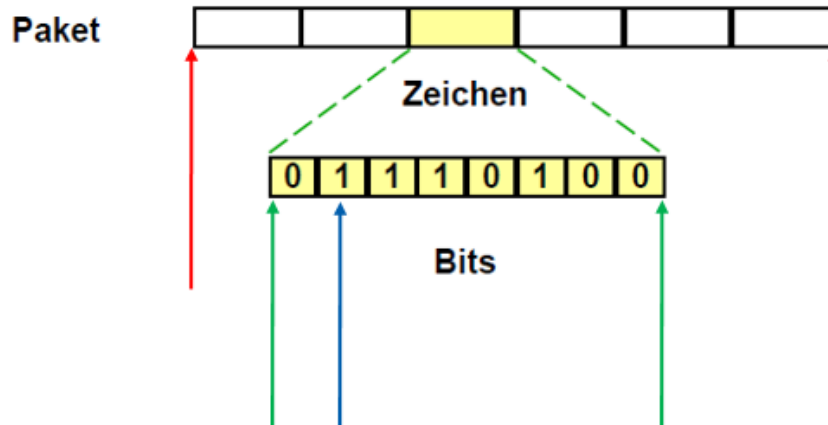
- Die bei einem seriellen Bus übertragene Bitfolge wird
 - Paket oder Rahmen (engl. Frame) genannt
 - ein Paket besteht aus mehreren Zeichen oder Bytes,
 - z.B. ASCII-Zeichen,
 - die ihrerseits dann in Einzelbits zerfallen



Synchronisation & Framing

Das Synchronisationsproblem gibt es für

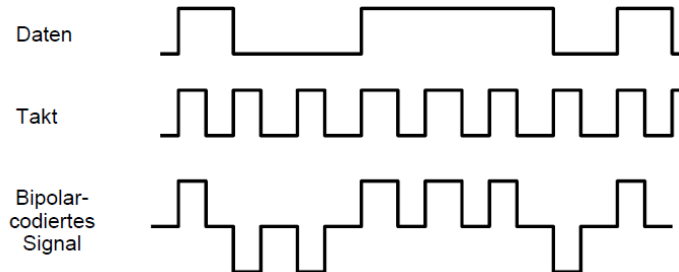
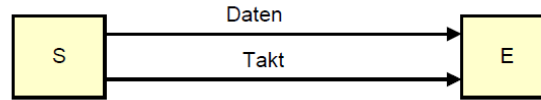
- das Erkennen von Anfang und Ende eines Pakets
- das Erkennen von Anfang und Ende eines Zeichens
- das Erkennen eines Einzelbits



Synchronisation & Framing

Die synchrone Datenübertragung:

- Sender und Empfänger synchronisieren ihre Takte für ein komplettes Paket.
→ Dazu beginnt jedes Pakets mit reservierten Zeichen zur Einsynchronisation.
- Mit Hilfe der übertragenen Information kann der Empfänger den eigenen Takt mit dem Takt des Senders synchronisieren.
- Alternative: Takt wird über separate Leitung übertragen (I²C)

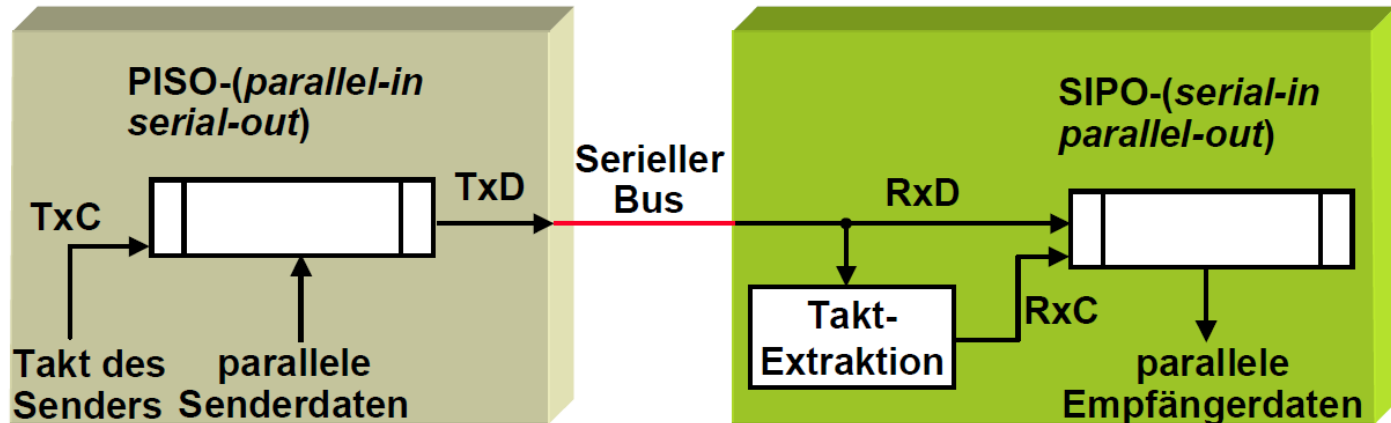
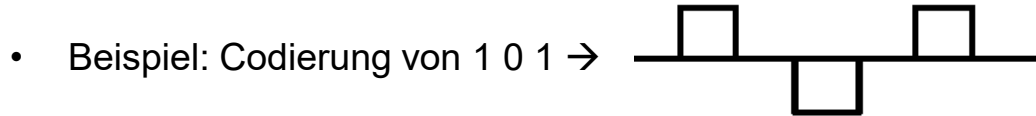


Synchronisation & Framing

Taktextraktion

Das Taktsignal kann beim Empfänger z.B. aus einem RZ- (*return-to-zero*)-Signal extrahiert werden:

- Bei einem RZ-Signal gibt es positive und negative Impulse für die Werte 1 und 0, daher sind drei Pegel notwendig.



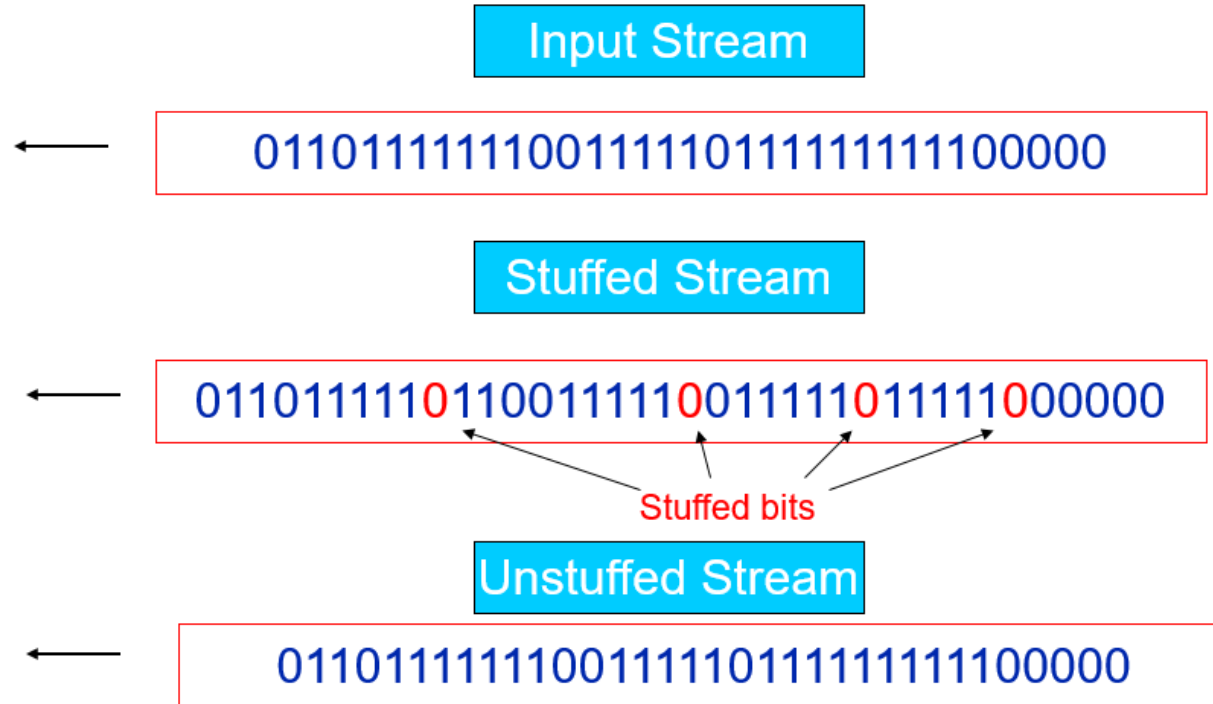
Synchronisation & Framing

Synchrone Übertragung

- Problem
 - Lange Telegrammstücke mit gleichem Pegel (NRZ) möglich!
- Abhilfe
 - Spezielle Startkennung
 - Definierte 1-0-Bitfolge
 - Bit Stuffing
 - Jedes n-te Bit kommt ein speziell eingefügtes Bit mit einem definierten Pegel
 - z.B. Bit Stuffing: 111111 \Rightarrow 1111101
 - Sender fügt außer bei den Flags nach jeder fünften ,1' eine ,0' ein und Empfänger löscht die nach fünf ,1' vorkommende ,0'.
 - Manchester-Codierung
 - Definierte Pegelwechsel

Synchronisation & Framing

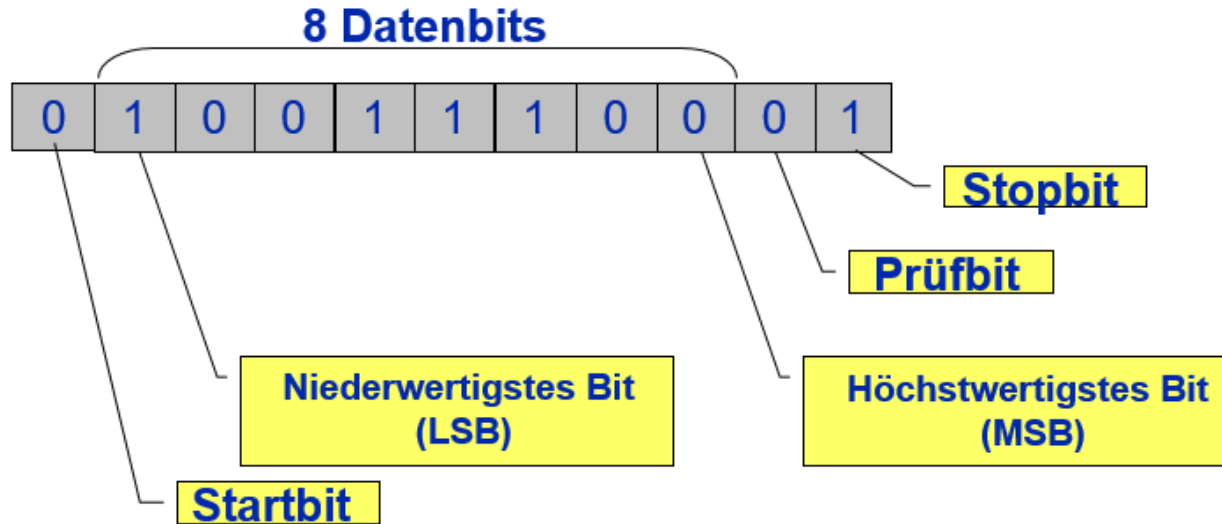
Bit Stuffing



Synchronisation & Framing

Asynchrone Übertragung

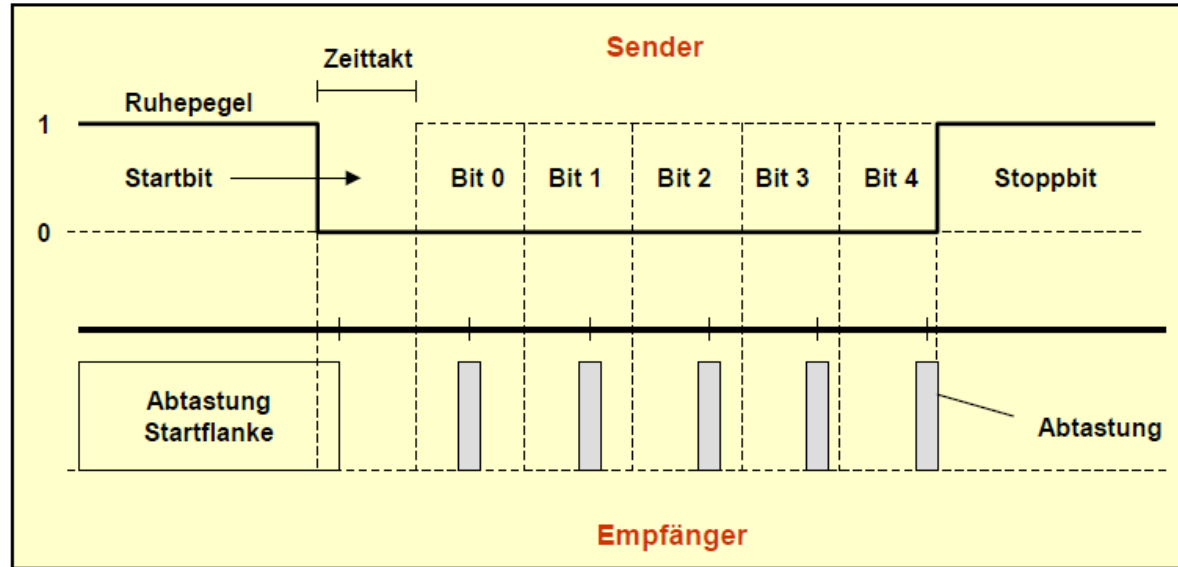
- Beispiel:
 - UART Format (Universal asynchronous receiver transmitter)



Synchronisation & Framing

Asynchrone Übertragung

- Timing:

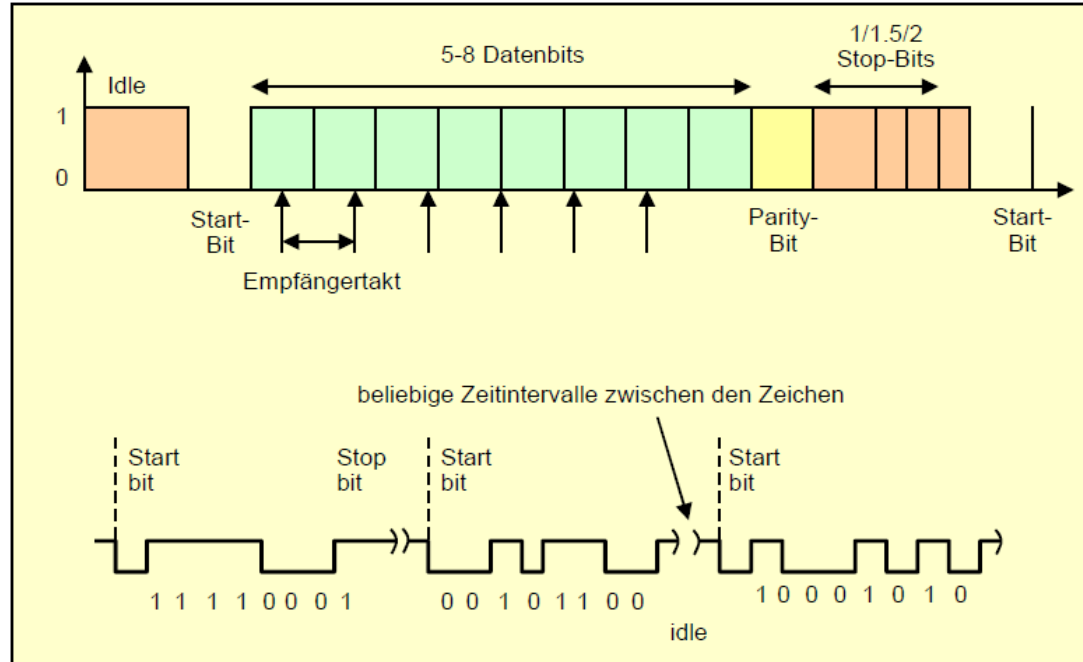


- Beschränkte Rahmengröße und Datenraten!!!**

Synchronisation & Framing

Asynchrone Übertragung

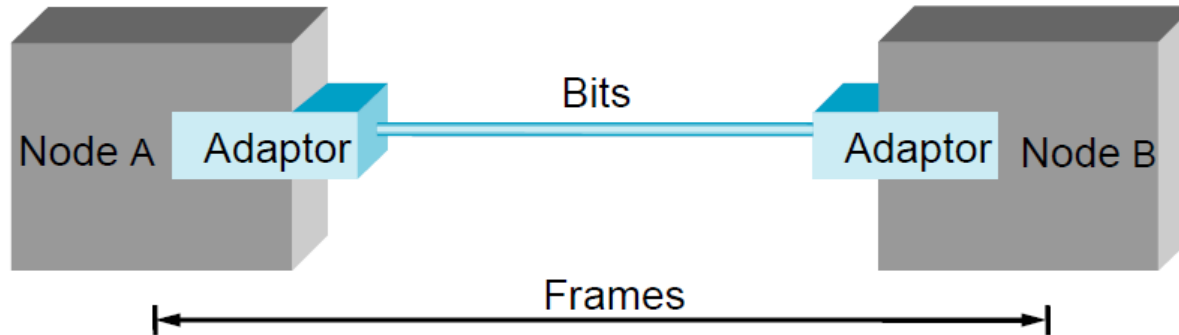
- Timing:



Synchronisation & Framing

Framing

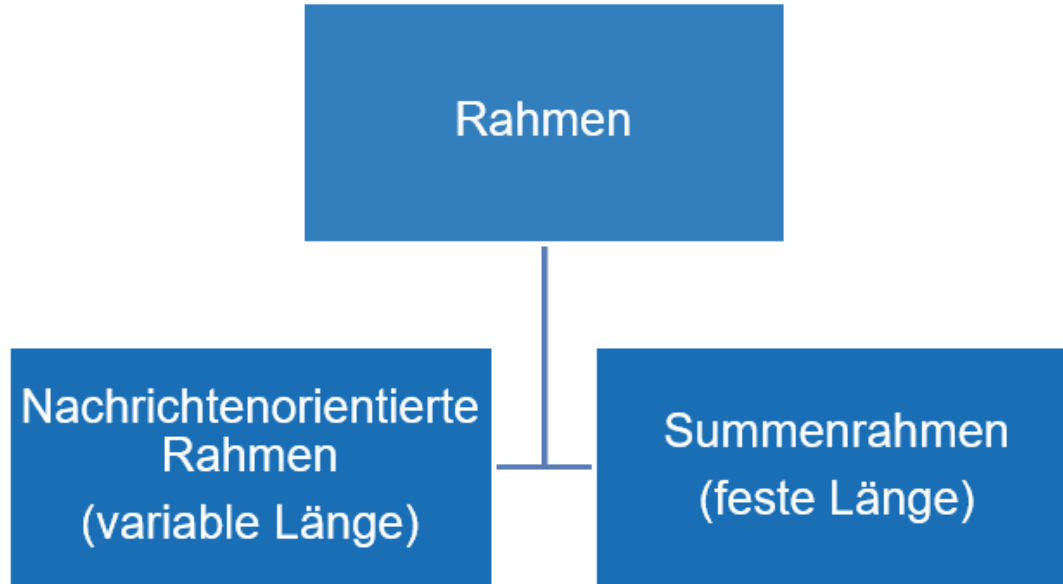
- Wir betrachten nun die Übertragung von Datenblöcken (hier: Frames) zwischen Rechnern:



- Zentrale Aufgabe:
 - Erkennung, wo ein Frame im Bitstrom anfängt und wo er aufhört
 - Framegrenzen müssen im Bitstrom erkennbar sein

Synchronisation & Framing

Framing



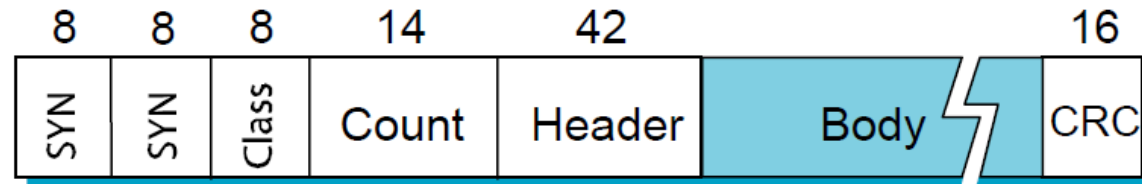
Synchronisation & Framing

Framing - Bytecounting

- **Byte-Count-Methode**

- Frameheader enthält Länge des Datenteils

- Beispiel:



Problem: Was passiert, wenn Byte-Count fehlerhaft übertragen wird?
→ Frame-Ende wird nicht korrekt erkannt

→ SYN-Zeichen am Beginn jedes Frames, um (wahrscheinlichen!) Anfang des folgenden Frames zu finden

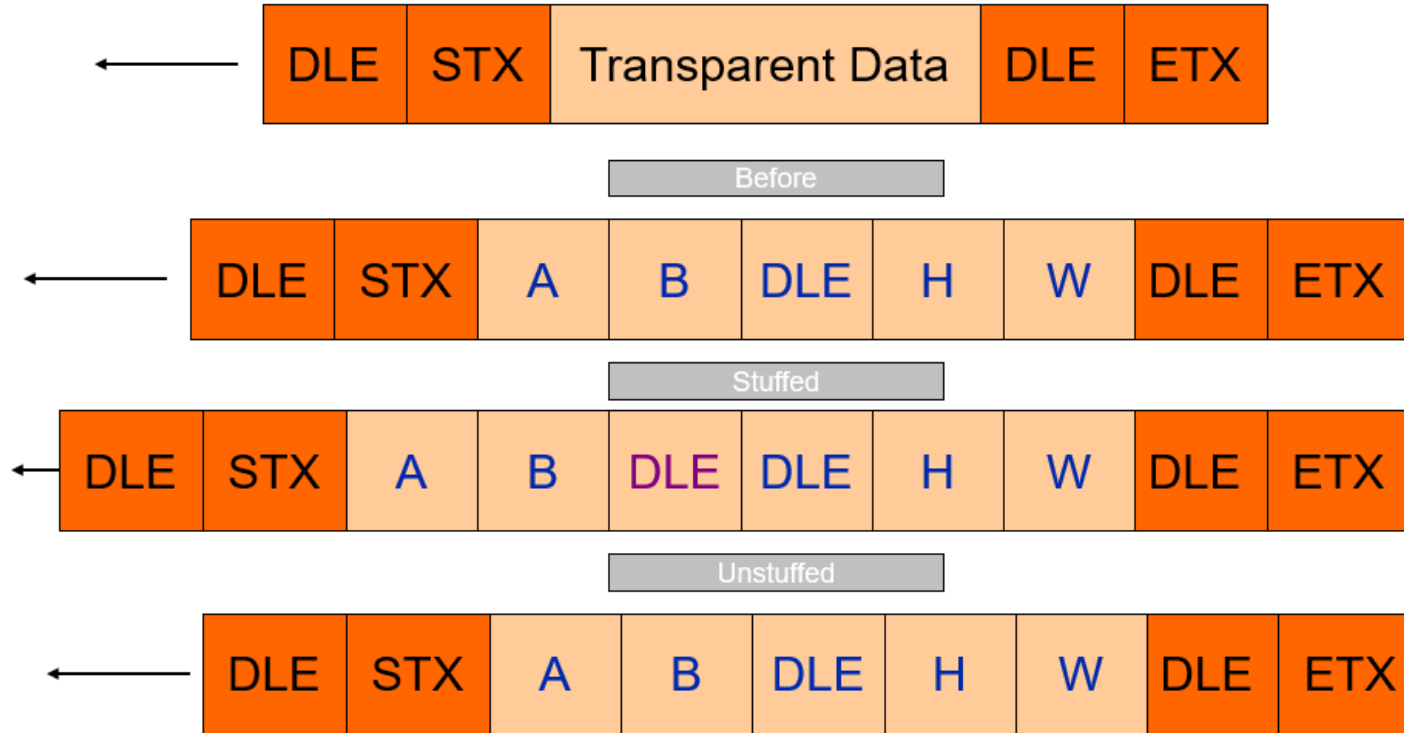
Synchronisation & Framing

Framing - FlagBytes

- FlagBytes:
 - Datenstrom wird als Zeichenfolge betrachtet.
 - Vereinbarung eines Start- (STX, start of text) und eines Endzeichens (ETX, end of text)
 - Beide Zeichen sind Spezialzeichen und werden sonst nicht benutzt.
 - **Problem** bei der Übertragung binärer Daten:
 - STX oder ETX können im Bitstrom vorkommen
 - **Ausweg:** Spezialzeichen DLE (data link escape)
 - Beginn des Pakets mit DLE, STX
 - Ende des Pakets mit DLE, ETX
 - dazwischenliegende Zeichen, die DLE bedeuten, werden beim Senden verdoppelt (byte stuffing) und beim Lesen wieder entfernt

Synchronisation & Framing

Framing - FlagBytes



Buszugriffsverfahren

Motivation

Wollen mehrere Teilnehmer an einem Bus gleichzeitig Daten übertragen,
müssen

- **mehrere Busse existieren** (teuer)

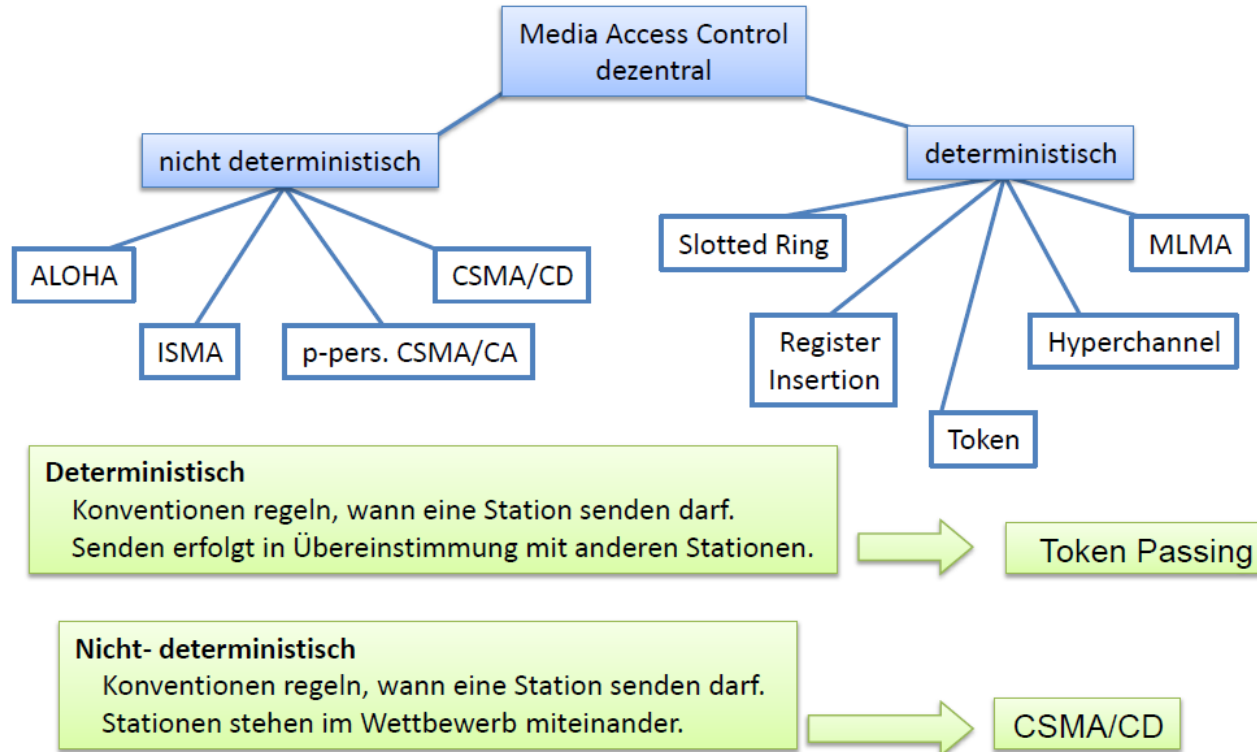
oder

- **der Zugriff muss serialisiert werden** → „Arbitrierung“

Hierbei finden verschiedene Buszugriffsverfahren Verwendung!

Buszugriffsverfahren

Klassifizierung von Buszugriffsverfahren



Buszugriffsverfahren

Klassifizierung von Buszugriffsverfahren

- **Deterministischer Buszugriff**
 - Festgelegtes Verfahren für Sendeberechtigung
 - Antwortzeitverhalten vorhersagbar

- **Nicht-deterministischer Buszugriff (Zufälliger Buszugriff)**
 - Permanentes Mithören
 - Ereignisgesteuerte Kommunikation
 - Niedrige mittlere Busbelastung
 - Antwortzeitverhalten nicht vorhersagbar

Buszugriffsverfahren

Klassifizierung von Buszugriffsverfahren

- **Zentrale Steuerung des Buszugriffes**
 - Die Steuerung über den Zugriff auf das Medium erfolgt von einem zentralen „Arbiter“ aus.

- **Dezentrale Steuerung des Buszugriffes**
 - Bei der dezentralen Bus-Arbitrierung versucht jede Komponente zunächst am Bus zu horchen, ob er gerade in Benutzung ist.
 - Wenn ja, wartet die Komponente, bis der Bus unbenutzt erscheint.

Buszugriffsverfahren

Master/Slave

➤ Master

- Aktiver Kommunikationspartner
- Gibt die Initiative für die Kommunikation
- Besitzt die “Busherrschaft”
- Sendet Nachrichten an passive Teilnehmer
- Sendet Anforderungen an passive Teilnehmer

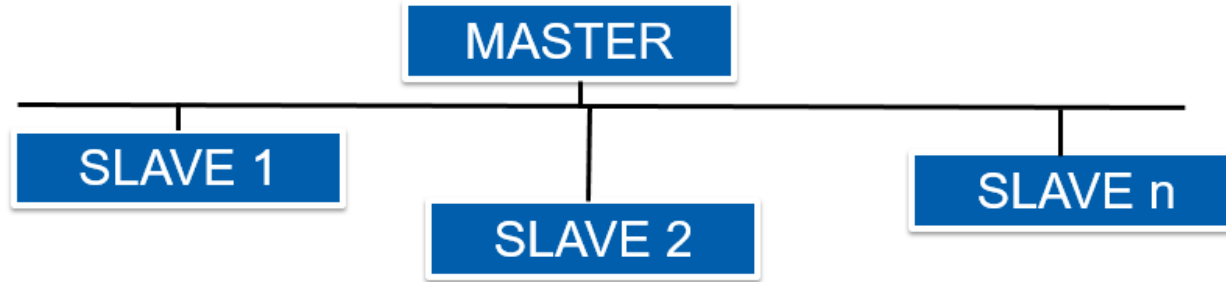
➤ Slave

- Passiver Kommunikationsteilnehmer
- Wird immer zur Kommunikation aufgefordert

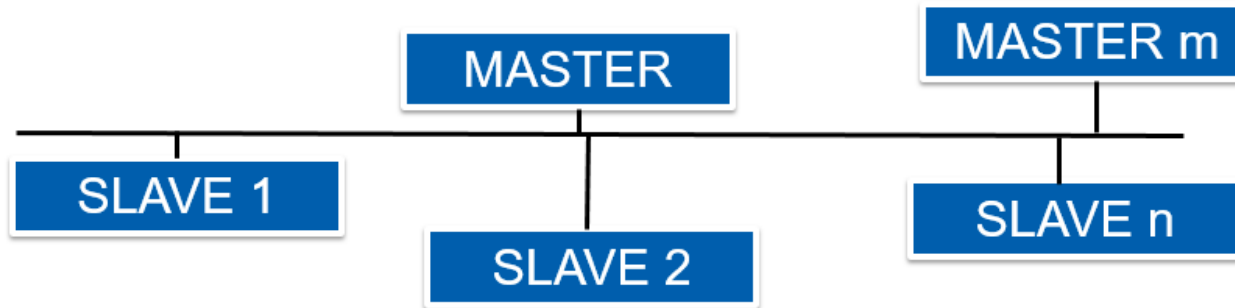
Buszugriffsverfahren

Single-Master/Multi-Master

- Single-Master-System



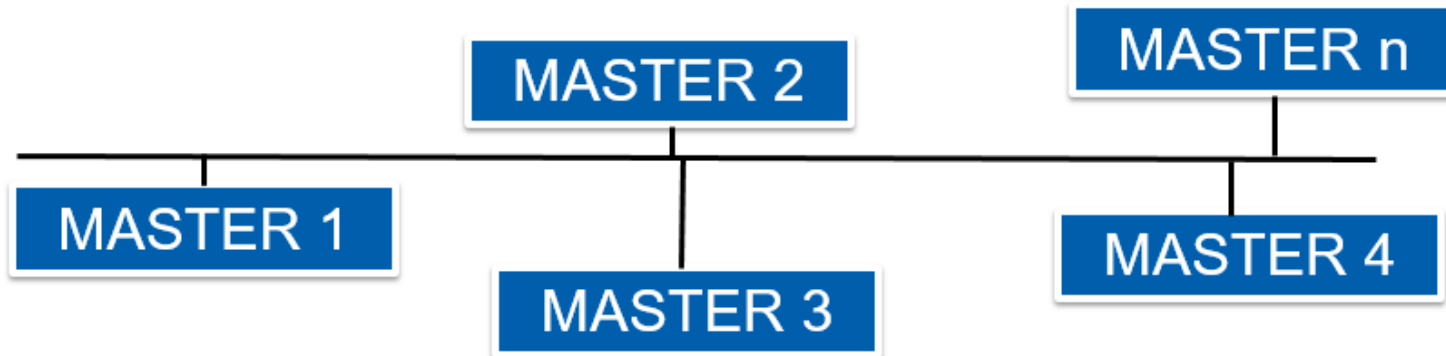
- Multi-Master-Multiple-Slave-System



Buszugriffsverfahren

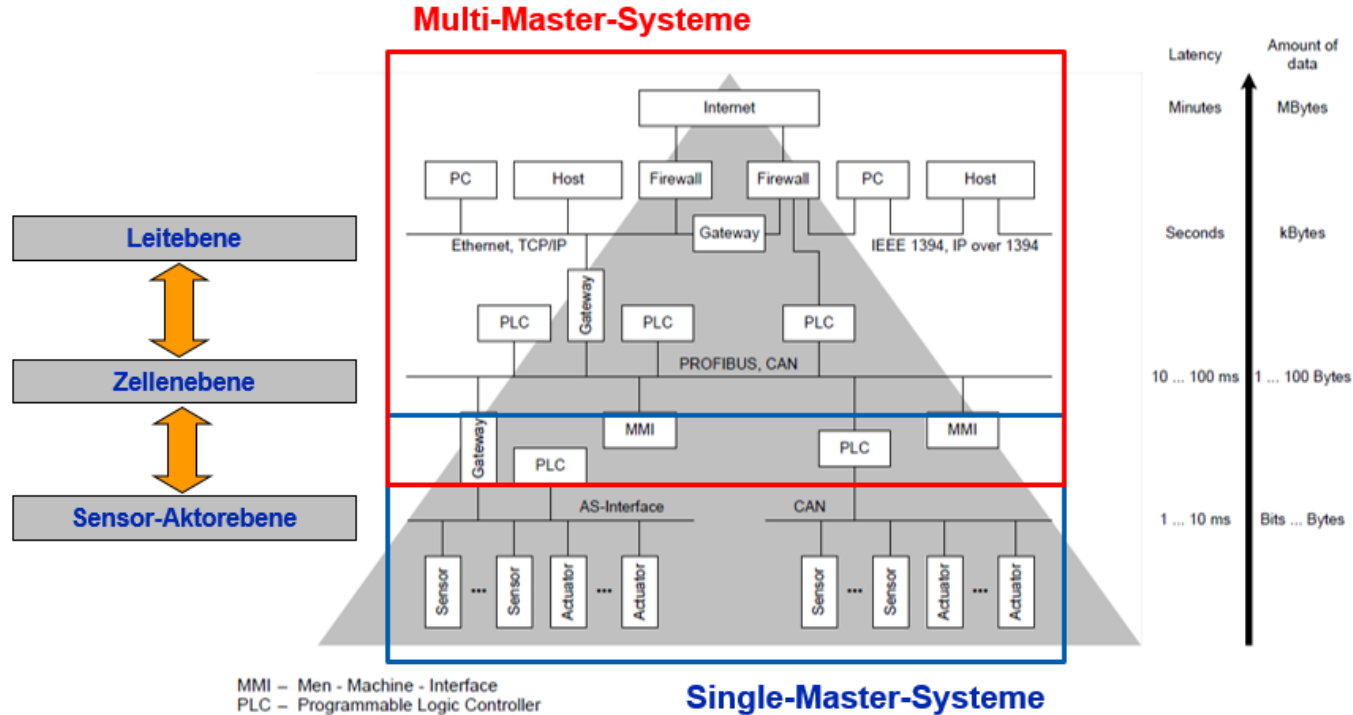
Single-Master/Multi-Master

- Multi-Master



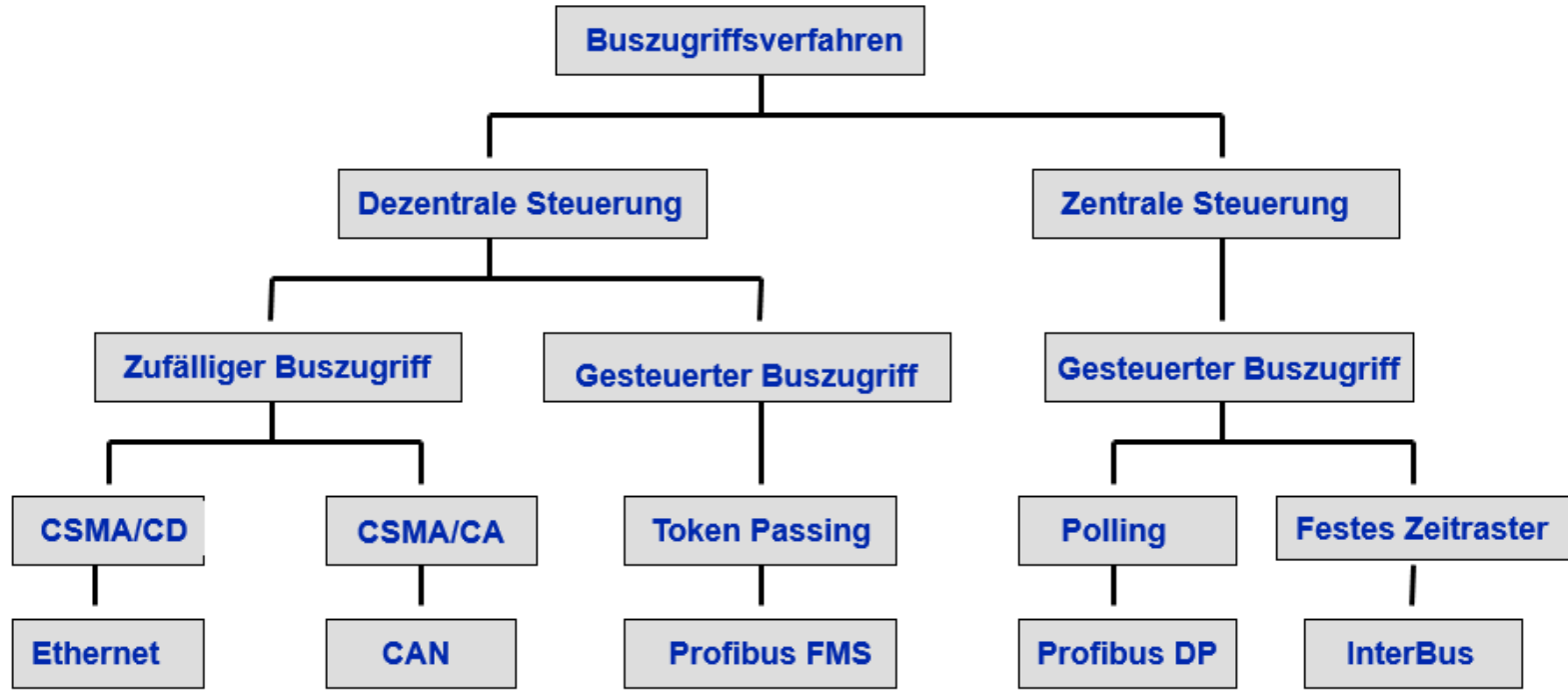
Buszugriffsverfahren

Single-Master/Multi-Master



Buszugriffsverfahren

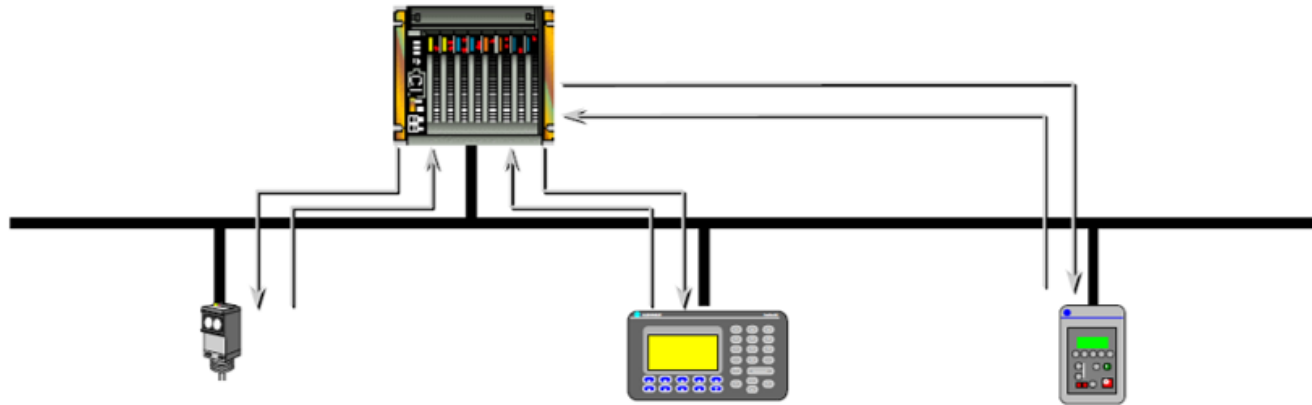
Übersicht der Verfahren



Buszugriffsverfahren

Master/Slave: Polling

- ❑ Gerät sendet auf Anforderung - meist des Masters - seine Daten (aktueller Status)
- ❑ Quelle-Ziel-Design / Master-Slave



Buszugriffsverfahren

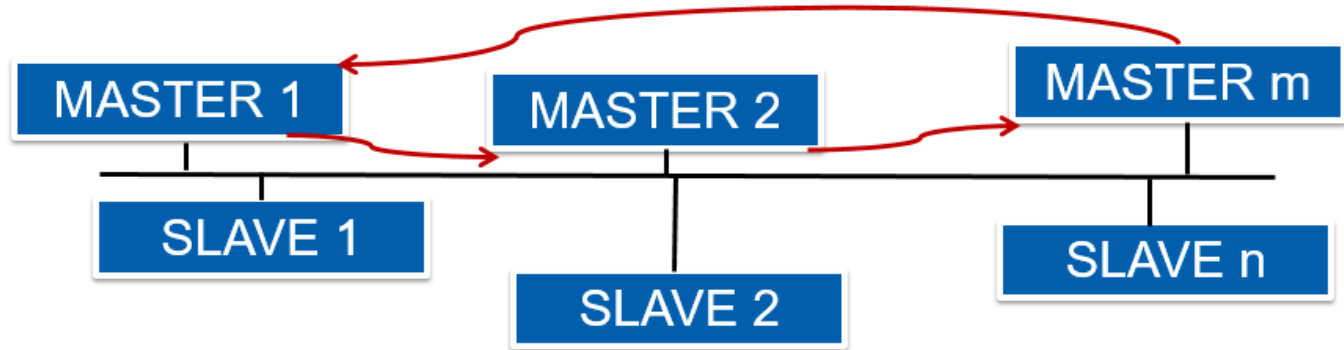
Master/Slave: Polling

- **Kommando-Antwort-Schema**
 - Master versendet Kommando an Slave
 - Slave antwortet, wenn er gefragt wird
 - Meistens zyklischer Betrieb
- **Vorteile**
 - „intelligente“ Master, „dumme“ Slaves
 - Echtzeitfähig, Zyklus ist genau berechenbar
- **Nachteile**
 - Wenn der Master tot ist, steht der gesamte Bus
 - Slave zu Slave Kommunikation nur über Master
 - Keine bedarfsorientierte Übertragung

Buszugriffsverfahren

Token-Passing → Token-Bus

Token-Bus:



Multi-Master-System:

- Es muss sichergestellt werden, dass nur ein Master die Zugriffsrechte hat,
- Ein TOKEN (Zugriffsrecht) wird an den nächsten Master weitergegeben, wenn der gerade aktive Master fertig ist.
- Das Token kann nur an den Nachbarn weitergeleitet werden → (logischer Ring)

Buszugriffsverfahren

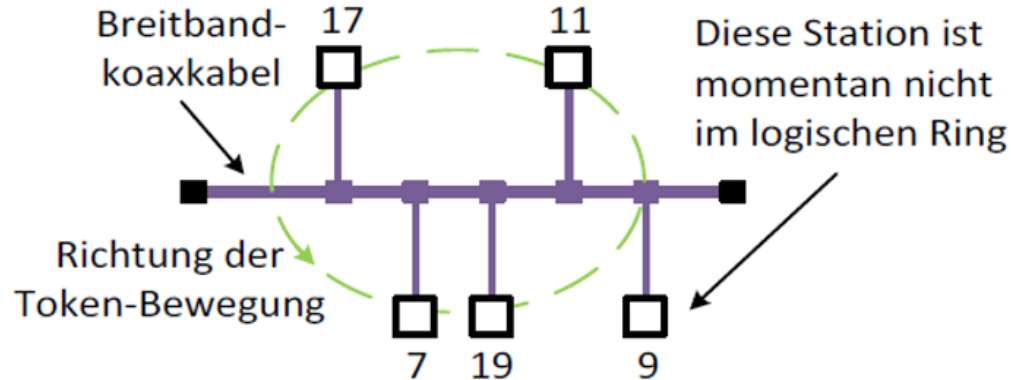
Token-Passing → Token-Bus

- Token wird über den Bus gesendet
- Steuerung ist recht komplex, da u.a. folgende Funktionen benötigt werden:
 - Initialisierung des log. Ringes nach Einschalten
 - Einfügen neuer Stationen in den log. Ring
 - Herausnehmen von Stationen
 - Verwaltung des Token →
 - Token darf nicht verloren gehen
 - Keine Existenz von zwei Token

Buszugriffsverfahren

Token-Passing → Token-Bus

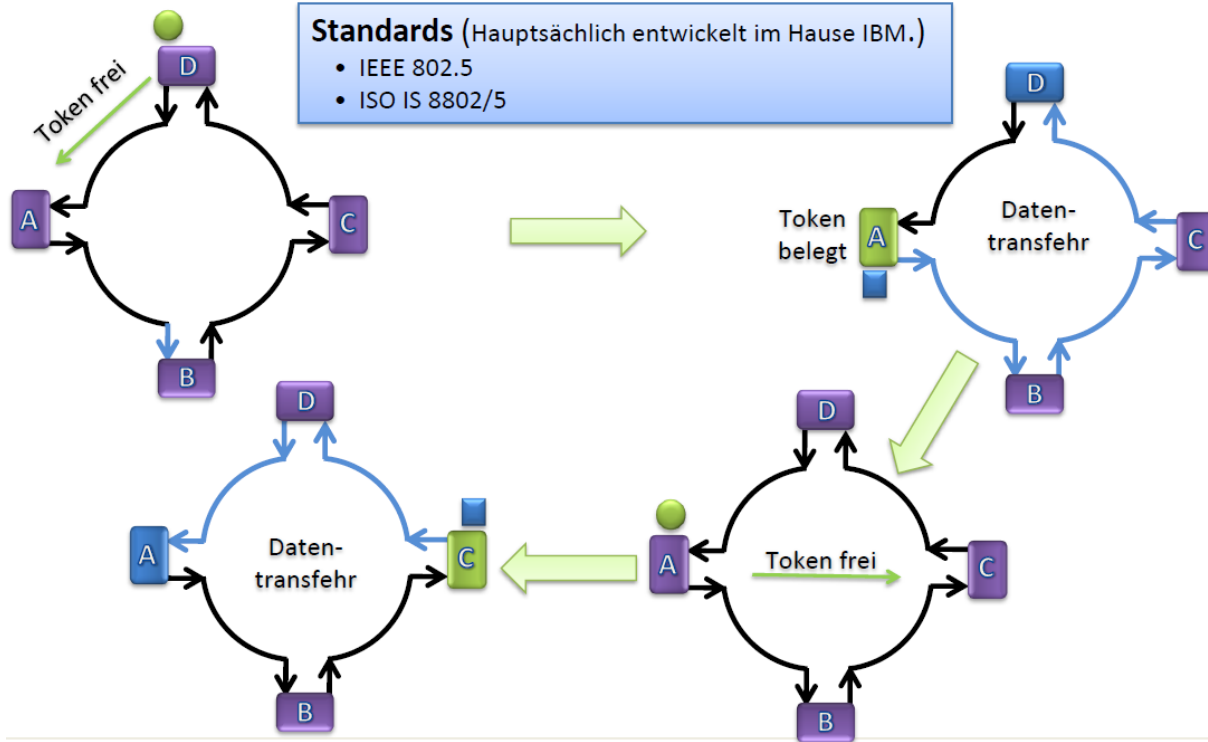
- Token-Bus in Multi-Master-System:



- z.B. IEEE 802.4

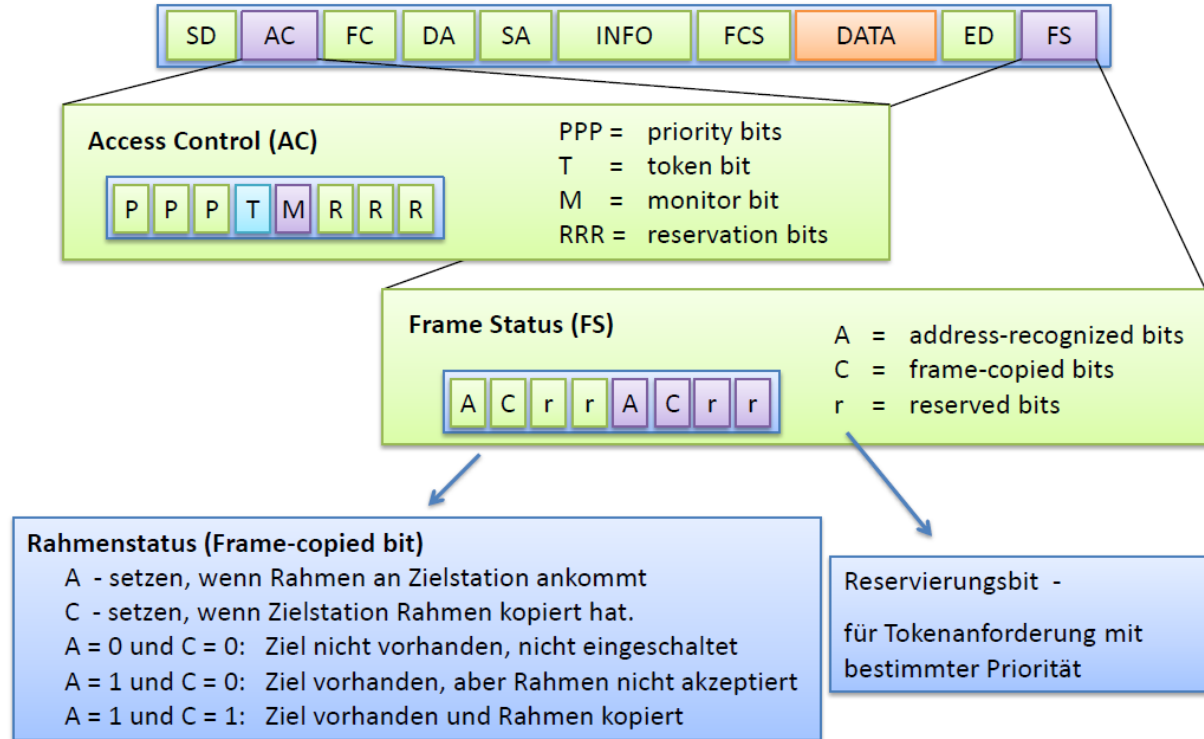
Buszugriffsverfahren

Token-Passing → Token-Ring



Buszugriffsverfahren

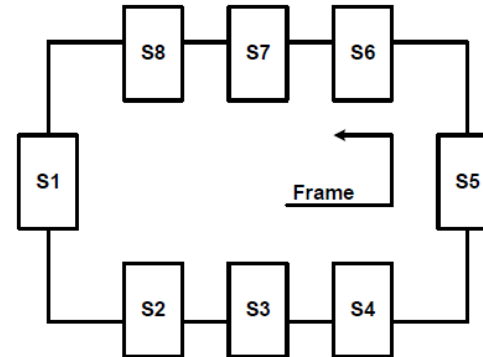
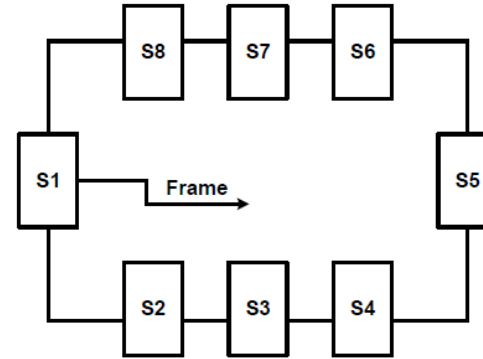
Token-Passing → Priorisierung



Buszugriffsverfahren

Token-Passing → Priorisierung

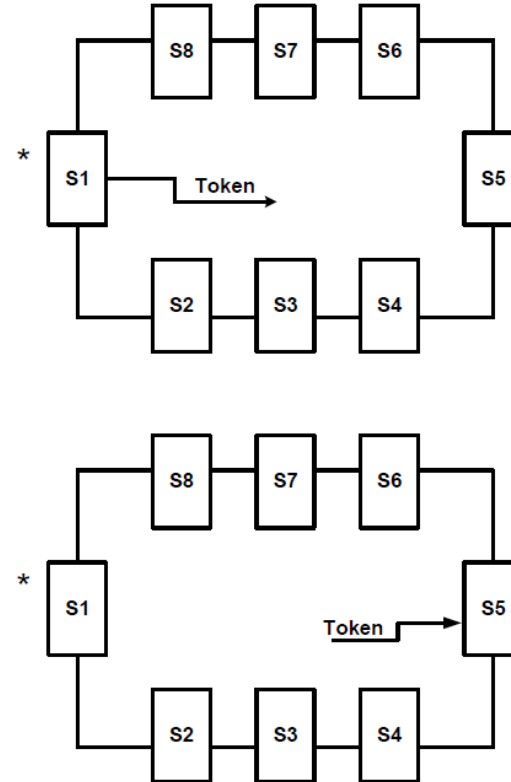
- Station S1 erhält das Token und überträgt ihren Rahmen mit normaler Priorität.
- S5 reserviert eine höhere Priorität im vorbeilaufenden Rahmen (RRR-Bits).



Buszugriffsverfahren

Token-Passing → Priorisierung

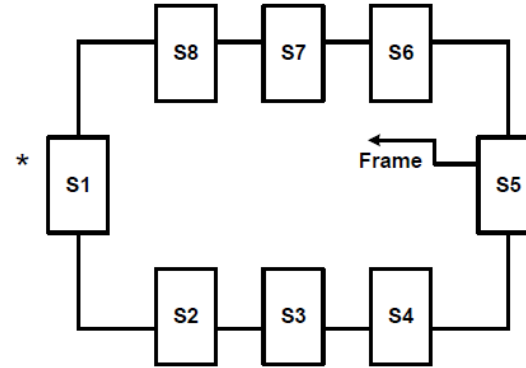
- S1 entfernt ihren Rahmen nach erfolgter Übertragung, erzeugt ein Token mit der von S5 reservierten Priorität und geht in den Zustand "priority-hold" über.
- S2, S3 und S4 haben keine Prioritätsrahmen, und das Token läuft weiter zu S5.



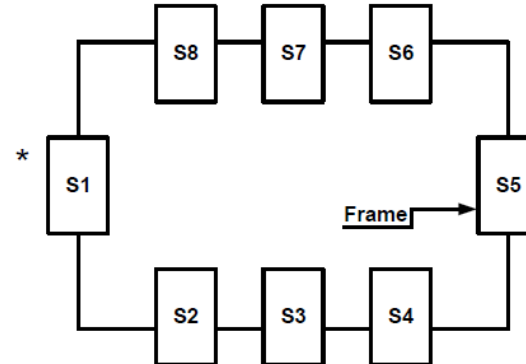
Buszugriffsverfahren

Token-Passing → Priorisierung

- S5 überträgt ihren Prioritäts-rahmen.



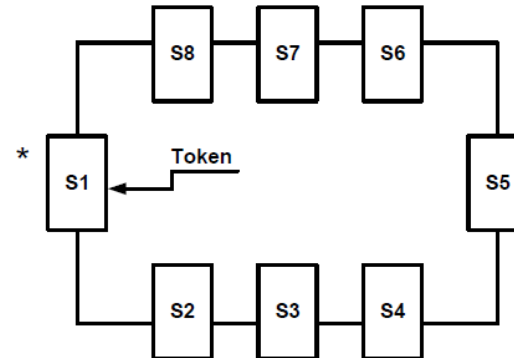
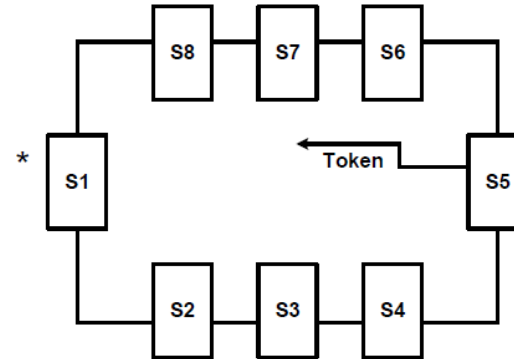
- Der Rahmen kommt zu S5 zurück.



Buszugriffsverfahren

Token-Passing → Priorisierung

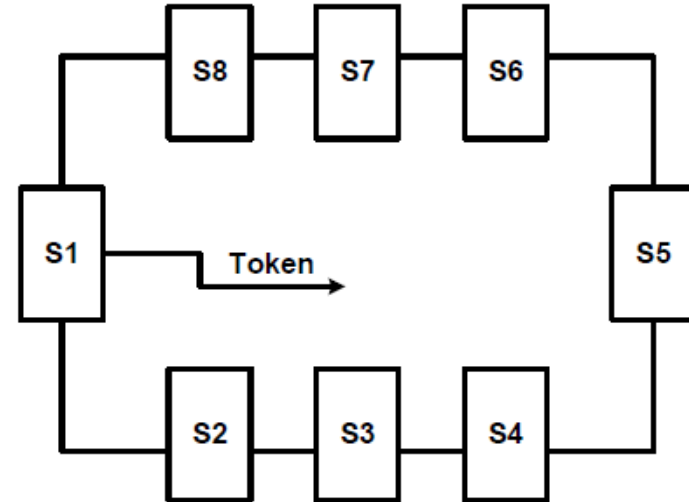
- S5 hat ihre Übertragung beendet und erzeugt ein Token mit der Priorität, die sie gerade benutzt hat (die höhere Priorität). S1, immer noch in "priority-hold", wartet auf ein Frei-Token mit dieser Priorität (die Priorität die S5 angefordert und S1 generiert hat).
- S1 empfängt das Frei-Token von S5 und erkennt die von ihr selbst erzeugte Priorität.



Buszugriffsverfahren

Token-Passing → Priorisierung

- S1 verlässt den Zustand "priority hold" (vorausgesetzt dass keine neue Prioritätsreservierung vorliegt) und erzeugt ein Token mit normaler Priorität. Falls S2 auf ein Token mit normaler Priorität wartet, kann sie jetzt mit der Übertragung beginnen.



Buszugriffsverfahren

Token-Passing → Übersicht

- **Zugriffsberechtigung wird mit Token erteilt**
 - Tokenbesitzer hat Bus für sich alleine
 - Token wird nach bestimmter Zeit automatisch weitergegeben
 - Zweifelsfälle regelt ein herausgehobener Master
- **Vorteile**
 - Slave-Slave Kommunikation möglich
 - Wer nichts senden will, gibt Token weiter
 - Bus kann auch nach Ausfall eines Master weiter betrieben werden
- **Nachteile**
 - Komplexe Verwaltung nötig
 - Tokenumlauf von der Anzahl Teilnehmer abhängig
 - Rekonfiguration, wenn neue Teilnehmer dazukommen

Ausblick

Weitere Buszugriffsverfahren...



UbiComp – Teil 6: Netzwerktechnik und industrielle Kommunikation III

Fragen?

Prof. Dr.-Ing. Dorothea Schwung